REMARKS and ARGUMENTS

Claims 1-24 are pending in the Application. All claims have been rejected.

Claim Rejections -35 USC § 103(a)

The Office Action has rejected Claims 1-5, 7-13, 15-21, and 23-24 under 35 U.S.C. § 103(a) as being unpatentable over Kadambi et al. (U.S. 6,934,830) (hereinafter "Kadambi") in view of Megiddo et al. (U.S. 6,996,676) (hereinafter "Megiddo"), in further view of Cherabuddi (U.S. 6,263,416) (hereinafter "Cherabuddi").

In addition, the Office Action has rejected Claims 6, 14 and 22 under 35 U.S.C. § 103(a) as being unpatentable over Kadambi et al. (U.S. 6,934,830) in view of Megiddo et al. (U.S. 6,996,676), in view of Cherabuddi (U.S. 6,263,416) and in further view of Choquette (U.S. 6,088,784) (hereinafter "Choquette").

However, the Office Action has failed to make a prima facie case of obviousness for the claims, and such rejections should be withdrawn.

The legal requirements for a prima facie case of obviousness are clear. "The examiner bears the initial burden of factually supporting any prima facie conclusion of obviousness." MPEP § 2142. It is well established that *prima facie* obviousness is only established when three basic criteria are met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991) (MPEP 2144).

The Office Action has failed to meet one or more of these requirements.

Cherabuddi discloses a central instruction window. Cherabuddi generally discloses known principles regarding the use of an instruction window to hold dependent instructions in an out-of-order machine. (Cherabuddi, Col. 1, lines62-65). A scheduling algorithm identifies the independent instructions within the instruction window, while the dependent instructions wait in the scheduling window until their operands have completed. (Cherabuddi, Col. 1, lines 1-14). Applicants do not dispute that Cherabuddi discloses the use of a scheduling window to hold waiting, future instructions until their operands have completed. Thus, the remainder of Applicants' remarks focus on the other cited references.

Kadambi discloses a system that includes a "register pane" which is smaller and faster than the register file and contains copies of a subset of registers from the register file. Kadambi simply teaches the use of a register pane, in which an operand is retrieved from the register pane instead of from the register file, when possible. (Kadambi, Abstract). Regarding placement of data INTO the register pane, Kadambi teaches a very simplistic approach. That is, Kadambi teaches that, if one or more of the operands for the instruction are not found within register pane 102, the missing operands are retrieved from register file 104 (step 308) and are stored into register pane 102 (step 310). (Kadambi, Col. 4, lines 34-39).

Kadambi further teaches that a number of different schemes can be used to determine which operands to OVERWRITE in the register pane, including LRU and round-robin. (Kadambi, Col. 4, lines 40-49). These are history-based replacement schemes (LRU, round-robin. There is no teaching in Kadambi toward a method or apparatus for tracking which values to place in the register pane based on the significance of the data to waiting, future instructions in the instruction stream.

Megiddo relates to caching of main memory pages (Col. 1, lines 17 – 22, and lines 31-39). Megiddo does not address or disclose register caches. For memory page caching, Megiddo discloses an adaptive replacement cache policy that dynamically maintains two lists of pages: recency list and frequency list. (Megiddo, Abstract). In the Background section, Megiddo discusses various replacement policies to determine which memory page is to be "paged out" if a cache is full. (Megiddo, Col. 1, lines 38-41). These methods include LRU, which captures locality but does not capture frequency. (See Megiddo, Col. 2, lines 7-10). Megiddo also discusses a least-recently-used replacement scheme (LRU), which replaces the least frequently used page. (Megiddo, Col. 2, lines 30-32). Megiddo also discloses a frequency-based replacement algorithm (FBR). The FBR maintains an LRU list, but also keeps a counter for each page in the cache. Page hits for pages in a middle or old section of an LRU list cause the counter to be incremented for that page. (Megiddo, Col. 3, lines 35-46). FBR is basically just a modified LRU scheme that is based on ***past*** page accesses and is not forwarding-looking.

Megiddo and Kadambi both discuss how information is chosen for EVICTION or replacement from a cache. They do not disclose, suggest or teach a selection algorithm for which information is to be put INTO the cache in the first place. Furthermore, they both utilize history-based eviction schemes, and do not provide for tracking which values to place in the register pane based on the importance of the data to waiting future instructions in the instruction stream.

Applicants disclose at least three opportunities for a physical register data to be written ***into*** a register cache: a demand miss, where physical register data is required, but is not contained by the register cache; physical register data may be produced by an execution unit; and

a prefetch mechanism may be employed to provide physical register data that may be required prior any miss penalty. (Application, para. 26).

Kadambi does not disclose any kind of selection algorithm for determining which information to place into the register pane when an opportunity such as a demand miss occurs. Instead, Kadambi discloses that missing operands for an issued instruction are always retrieved from the register file and are stored in the register pane. See Kadambi, Fig. 3, 308 and Col. 4, lines 36-39. Similarly, Megiddo discloses "demand paging" in which a page is copied from the auxiliary memory to the cache any time a requested memory page is not found in the cache. (Megiddo, col. 1, lines 31-39).

In contrast, for each opportunity to write information INTO the register cache, Applicants disclose a scheme for determining whether to actually go ahead and place physical register data into the register cache at that opportunity. Applicants disclose a count table that may be accessed to determine a potential significance when an opportunity to cache occurs. For at least one embodiment, "the insertion of the data may be conditional on the potential significance being high..." (Application, para. 30). By selectively caching certain physical register data (e.g., those predicted to have a potentially high significance) in the register cache, register read activity may be filtered from the register file. (Application, para. 22).

Note that this selection of data to be written into the cache is NOT the same thing as selecting which data to evict from a cache. That is, embodiments of the invention relate to "predictive filtering of register cache entries" to determine which entries go INTO the register cache. (See 312, Fig. 3; Application, para. 28). Replacement schemes do not read on this predictive filtering.

Applicants disclose that, in at least one embodiment, the determination of whether a physical register should be entered into the register cache is based on at least two factors. First, those physical register references that are part of an instruction window are tracked. (Application, para. 23 – "The CLT 104 may track physical register references that are ***currently active in the instruction window,*** emphasis added).

Importantly, the instructions in the instruction window may include waiting, future instructions that have not yet executed. For example, the Specification teaches "large instruction windows for extracting ILP in an out-of-order execution core" (Application, para. 22). The Specification also teaches that the instruction window may be coupled to a reservation station that may buffer operands for the waiting instructions that are in the instruction window: "The reservation station 114 may buffer operands of instructions waiting to be scheduled for execution or issue." (Application, para. 23).

The fact that an instruction window may include waiting, future instructions is not only supported by the Specification, but is also well-known in the art. For example, The Examiner has stated in the Office Action that use of a central instruction window allows for instructions to execute out-of-order and speculatively. (Office Action, page 4). One of skill in the art will thus recognize that those instructions that are currently active in the instruction window include instructions that have not yet been executed but are awaiting for the operands on which they depend to become available. (See, e.g., discussion of instruction windows at Col. 2, lines 1-14 in the Cherabuddi reference cited in the Office Action). Thus, an instruction window includes ***future*** instructions that have not yet executed but are waiting for their operands to become available. The potential significance that drives the determination of whether or not to place physical register data into the register cache is therefore based on references to the physical

register. It is not history-based but is forward-looking – it is based on references in future, waiting instructions in the instruction window.

Second, the significance for a physical register indicates the number of times that the physical register is referenced by instructions in the instruction window. That is, the significance is higher for a physical register that is likely to be referenced relatively often in future instructions. Applicants disclose, for instance, that a counter 204 may be used to indicate the potential significance of placing data into the register cache. (Application, para. 24). The counter 204 maintains, for its associated physical register, a reference count of physical register accesses to the physical register in the instruction window. (Id.) The counter for each physical register in the CLT may be incremented for each of the physical register references in the register file. (Application, para. 32). "The counter 204 may be indicative of the potential significance of placing the data into the register cache." (Id.)

Claim 1. Claim 1 recites, in part, "entering said data into ~~updating~~ at least one register cache according to said potential significance." (amended Claim 1, in part). None of the cited references suggest, teach or disclose this element.

The Office Action states that one of skill in the art would be motivated to use the FBR algorithm to replace registers that won't be used. However, applicants claim entering data into at least one register cache according to said potential significance. Replacement schemes do not address this limitation. Thus, a prima facie case has not been made out and Claim 1 is allowable.

Claim 1 also recites, in part, "utilizing said reference count to determine the potential significance of data in said physical register to future instructions" (amended Claim 1, in part). None of the cited references suggest, teach or disclose this forward-looking element. None of

the references disclose a scheme to determine significance of data utilizing a reference count that is based on physical register references in a *waiting* instruction.

Because none of reference individually teaches the recited elements, their combination is insufficient to make a prima facie case of obviousness. Thus, a prima facie case has not been made out and Claim 1 is allowable. In addition, Claims 2-8, which depend from Claim 1, are also allowable for at least the foregoing reasons.

Claim 9. Claim 9 recites, in part, "reference count circuit is to enter data of ~~updates~~ said at least one register cache according to said potential significance." (amended Claim 9, in part). None of the cited references suggest, teach or disclose this element.

The Office Action states that one of skill in the art would be motivated to use the FBR algorithm to replace registers that won't be used. However, applicants claim entering data into at least one register cache according to said potential significance. Replacement schemes do not address this limitation. Thus, a prima facie case has not been made out and Claim 9 is allowable.

Claim 9 also recites, in part, "an instruction window to hold the plurality of instructions, the plurality of instructions including at least one waiting instruction whose data is not yet available in the physical registers" and "a counter look-up table to track the number of references to each of said one or more physical registers in the instruction window" (amended Claim 9, in part). None of the cited references suggest, teach or disclose these forward-looking elements. None of the references disclose a scheme to track the number of references to physical register references for *waiting* instructions in an instruction window.

Because none of reference individually teaches the recited elements, their combination is insufficient to make a prima facie case of obviousness. Thus, a prima facie case has not been

made out and Claim 9 is allowable. In addition, Claims 10-16, which depend from Claim 9, are also allowable for at least the foregoing reasons.

Claim 17. Claim 17, recites, in part: "a processor including an instruction window to hold a plurality of instructions, the plurality to include ~~of~~ at least one future instruction, one or more physical registers to store data associated with said at least one future instruction, a counter look-up table to track one or more physical register references in the instruction window ~~associated with~~ to said one or more physical registers, a reference count circuit to determine ~~a reference count for said one or more physical registers and~~ a potential significance for data of said one or more physical registers based on said physical register references" (amended Claim 17, in part). None of the cited references suggest, teach or disclose this element.

As claimed in Claim 17, the look-up table tracks one or more physical register references in the instruction window (which includes at least one future instruction) and determines a potential significance of physical register data based on these physical register references. That is, the determination of potential significance is forward-looking and is based on physical register references of at least one future instruction in the instruction window. None of the references disclose, suggest or teach these limitations. Thus, a prima facie case has not been made out and Claim 17 is allowable.

Claim 17 also recites, in part, "reference count circuit ~~updates~~ is to enter data into said at least one register cache according to said potential significance" (amended Claim 17, in part). The Office Action states that one of skill in the art would be motivated to use the FBR algorithm to replace registers that won't be used. However, applicants claim entering data into at least one

register cache according to said potential significance. Replacement schemes do not address this limitation. Thus, a prima facie case has not been made out and Claim 17 is allowable

Because none of reference individually teaches the recited elements, their combination is insufficient to make a prima facie case of obviousness. Thus, a prima facie case has not been made out and Claim 17 is allowable. In addition, Claims 18-24, which depend from Claim 17, are also allowable for at least the foregoing reasons.

Accordingly, Applicants respectfully submit that the applicable rejections have been overcome and must all be withdrawn. Applicants reserve all rights with respect to the application of the doctrine equivalents. Applicant respectfully requests that a timely Notice of Allowance be issued in this case. If the Examiner feels that an interview would help to resolve any remaining issues in the case, the Examiner is invited to contact Shireen Bacon of Intel, at (512) 732-3917.

Please charge any shortages and credit any overcharges to our Deposit Account No. 02-2666.

Respectfully submitted,

Dated: August 7, 2006

/Shireen Irani Bacon/

Shireen Irani Bacon
Reg. No. 40,494
Tel.:(512) 732-3917

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026